

MIDI Converter Ver.-1.30

Rex Sabio
242

Table of Contents

Introduction	2
1. Midi2Bin32X	3
2. SeConv32X	8
3. Midi2Txt32X	10
MIDI Data Format Tables	11

Introduction

The Midi Converter converts standard MIDI files, prepared using Tone Editor 32X, into 32X Sound Simulator data files according to the data format for SUPER 32X sound drivers. There are three types of Midi Converters:

- Midi2Bin32X

This package converts a given standard MIDI file into a binary file for use with the BGM.

The converted file can be used as sequence data for the 32X Sound Simulator.

- SeConv32X

This package converts a given standard MIDI file into a binary file for use with the SE.

The converted file can be used as sequence data for the 32X Sound Simulator.

- Midi2Txt32X

This package converts a given standard MIDI file into a text file for use with the BGM.

You can use Midi2Txt32X for editing data as text. For details on this data format, see the "68000 Sound Driver Manual".

Rex Sample
242

1. Midi2Bin32X

Conversion Procedures

1. <Creating BGM data>

As a first step, create a standard MIDI file for the BGM.

Create the data that can be properly played using the Tone Editor. Save the data in the standard MIDI file format.

Save standard MIDI files in Format 1.

Use the following fixed tracks for the sequencer, leaving unused channels as vacant tracks:

Track 1	PCM0
Track 2	PCM1
Track 3	FM0
Track 4	FM1
Track 5	FM2
Track 6	FM3
Track 7	FM4
Track 8	FM5
Track 9	PSG0
Track 10	PSG1
Track 11	PSG2
Track 12	PSG NOISE
Track 13	PWM0
Track 14	PWM1
Track 15	PWM2
Track 16	PWM3

(A temporary track is automatically created before Track 1.)

Note: Allocate a PCM track for each PCM channel used. Delete any unused tracks. If PCM is not used, Track 1 becomes FM1.

Note: When using an FM DRUM KIT, be sure to use an FM6 track.

Note: Be sure to allocate enough tracks for the data to be created.

Note: Be sure to observe the cautionary notes given below.

- Setting a loop

To set a loop, use a control change.

	Control Number	Parameter
Beginning of a loop	98	64
End of a loop	98	65

- Setting the Qsound

Use a control change to specify the on/off actions for the Qsound:

	Control Number	Parameter
Qsound on	29	A value from 0 through 63
Qsound off	29	A value from 64 through 127

Use a control change pan-pot to specify a pan-pot for the Qsound.
(Use the sequencer's pan-pot control as is.)

Note: Using the Qsound requires a separate sub-license agreement and a sound driver specific for the Qsound.

2. <Creating data using the Tone Editor 32X>

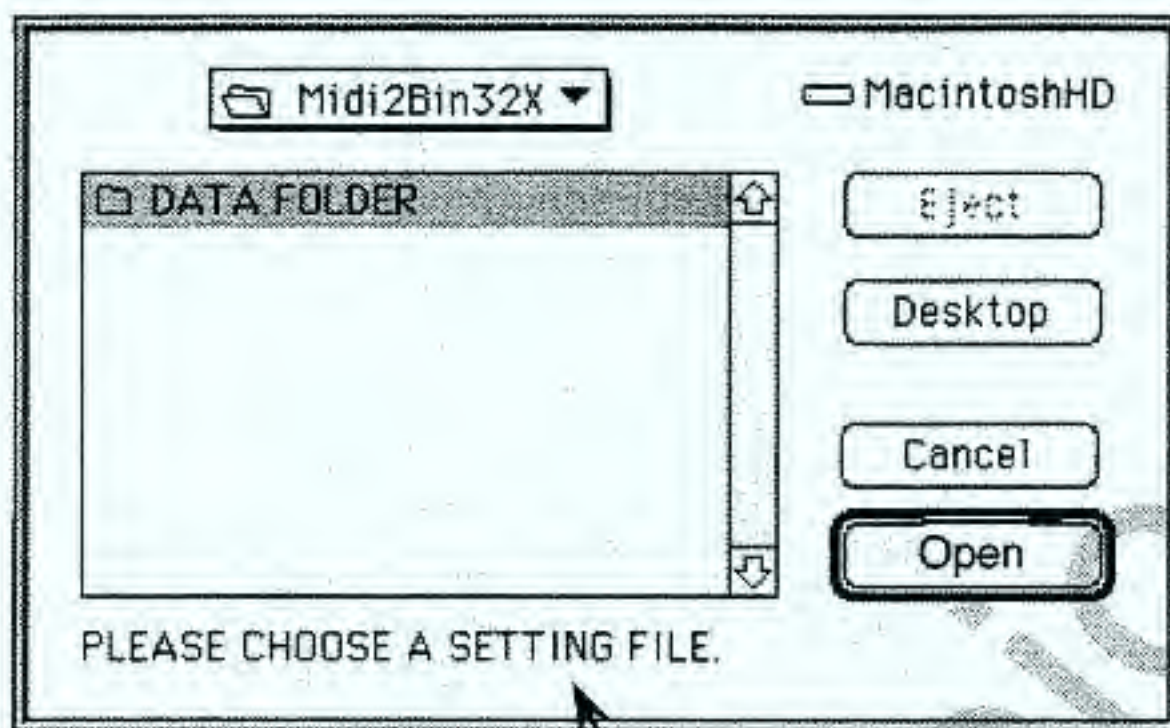
As the next step, use the Tone Editor 32X to create a MIDI setting file. To create this file, complete the MIDI-setting process using the Tone Editor 32X, and save the results. When using a velocity table, create a file for the table. The table value should be 64 or less.

When a velocity table is used, a maximum velocity value of 127 can be allowed in the sequencer. For other velocity types, however, the maximum allowable velocity value in the sequencer is 64. For details on the MIDI setting and velocity tables, see the "Tone Editor 32X Manual".

3. <Conversion>

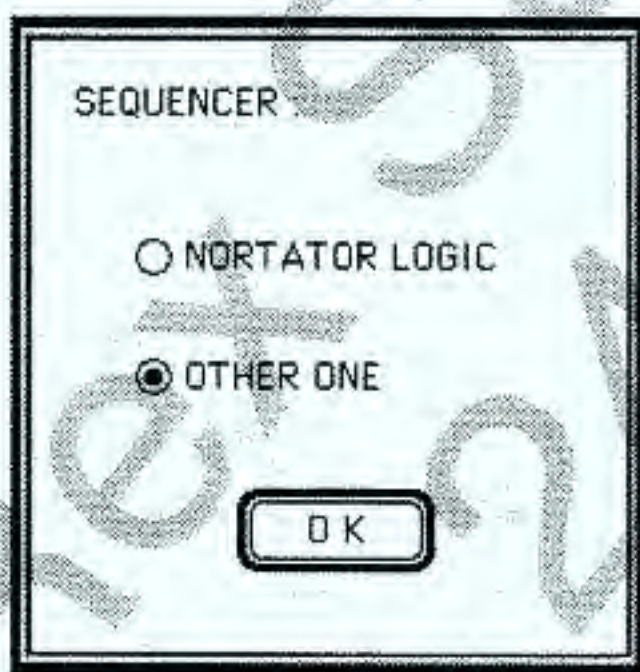
Double-click on MIDI2BIN32X. Perform the conversion process by following the guidance information that is displayed on the screen.

- Selecting a file



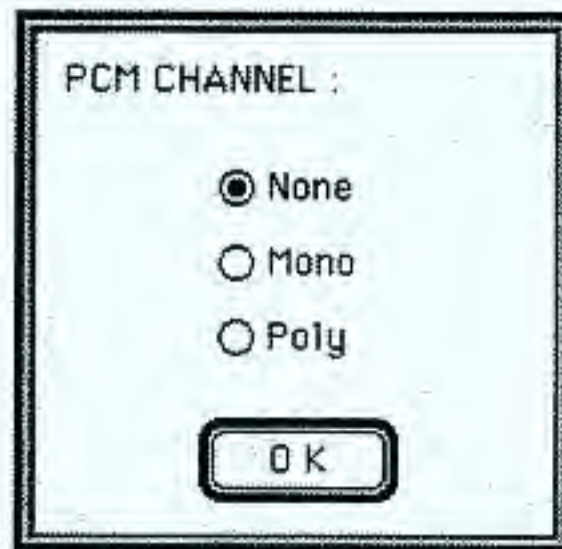
Select the desired file according to the indicated instruction.

- Selecting a sequencer



When creating a standard MIDI file, specify whether Notator LOGIC or another sequencer is used (see "Creating BGM Data").

- Specifying the number of PCM channels

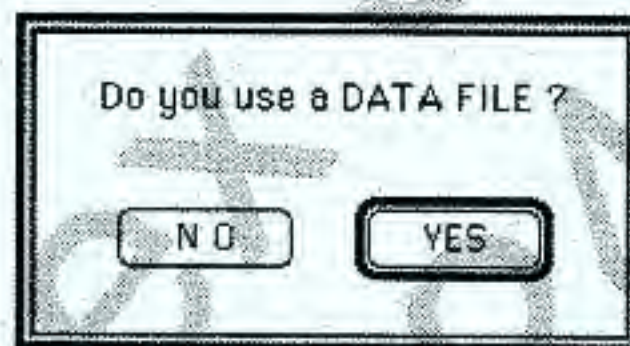


Specify the number of PCM channels in which PCM will be used.

None ... None
Mono ... One channel
Poly ... Two channels

Note that varying the number of PCM channels changes the way in which sequencer tracks are used.

- Selecting a data file



YES ... Add the following for each piece of music:

- FM timbre data
- Table envelope data
- Table vibrato data
- FM drum kit data

NO ... Use one file for each type of data.

Normally, select the "NO" option.

Notes:

* When using the System Exclusive, save the MIDI channel for all tracks for the sequencer by setting the MIDI channel to 1 (thus creating a standard MIDI

file). If this condition is not met, the System Exclusive separates into another track and prevents the converter from working properly.

* When using the Vision, use another sequencer to create a standard MIDI file. If musical scale data and program changes exist at the same time, the musical scale data always floats to the top, and this causes delta time to be 0.

* When using Notator LOGIC, be sure that a blank space (to be distinguished from the pause note [KYUUFU - UNKNOWN TERM -- "PAUSE NOTE" IS AN UNEDUCATED GUESS] is not inserted at the beginning of the score. When the standard MIDI file is converted, the track in which a blank space occurs is automatically moved to the beginning of the music. Be sure that there are no empty tracks. If there are empty tracks, use either a program change or a control change to insert dummy data in those tracks. If a track is completely empty, the track cannot be ignored when a standard MIDI file (SMF) is saved, thus causing the track setting to be misaligned.

* The Midi Converter does not support the following Control Change for the Tone Editor 32X:

- Hard LFO

* The Midi Converter does not support the following System Exclusive for the Tone Editor 32X:

- Sound effect-mode setting
- Direct manipulation of FM sound sources
- Detuning specification in units of division ratios

* Control changes are described in the section on "MIDI Data Format Tables".

2. SeConv32X

Conversion Procedures

1. <Creating SE data>

As a first step, create a standard MIDI file for the SE.

Create the data that can be properly played using the Tone Editor. Save the data in the standard MIDI file format.

Save standard MIDI files in Format 1.

The sequencer allocates tracks for the number of channels to be used. In contrast to the BGM processing, the sequencer does not use fixed channels (do not allocate excess tracks).

2. <Creating data using the Tone Editor 32X>

As the next step, use the Tone Editor 32X to create a MIDI setting file, as in the case of Midi2Bin32X.

3. <Conversion>

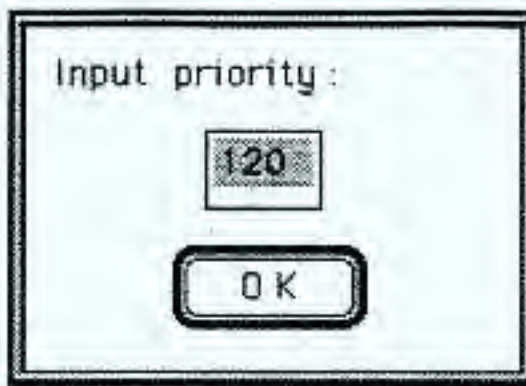
The SeConv32X conversion procedure largely parallels the Midi2Bin32X conversion procedure. The following describes only those parts of the conversion procedure that differ from the Midi2Bin32X conversion procedure:

- Setting the channel to be used

CHANNEL SELECT :		TRACK 1	
<input checked="" type="radio"/> PCM0	<input type="radio"/> FM0	<input type="radio"/> PSG0	<input type="radio"/> PWM0
<input type="radio"/> PCM1	<input type="radio"/> FM1	<input type="radio"/> PSG1	<input type="radio"/> PWM1
	<input type="radio"/> FM2	<input type="radio"/> PSG2	<input type="radio"/> PWM2
	<input type="radio"/> FM3	<input type="radio"/> PSG3	<input type="radio"/> PWM3
	<input type="radio"/> FM4		
	<input type="radio"/> FM5		
<input type="button" value="OK"/>			

Select the channel to be allocated, beginning with track 1 of the sequencer. The dialog box disappears when the required number of tracks has been selected.

- Determining the priority



On this dialog box, select the priority for the sound effect. Priority is specified using a decimal number 0 through 255. The larger the value, the higher the priority.

Note:
See the notes on MIDI2BIN32X.

Rex Sabio
242

3. Midi2Txt32X

Conversion Procedures

1. <Creating BGM data>

As in the case of Midi2Bin32X, create a standard MIDI file. The number of tracks should not be changed according to the number of PCM channels used.

2. <Creating data using the Tone Editor 32X>

Create data as in the case of Midi2Bin32X.

3. <Conversion>

1. Double-click on Midi2Bin32X.

2. When a dialog box appears, select the sequencer in which a standard MIDI file was created.

3. When using the MIDI setting file for the Tone Editor 32X, check the "User Preferences" item in the "Setting" menu bar.

4. Select the desired file according to the instructions given in the dialog box. See the section on "Midi2Bin32X".

5. The application terminates when all the conversion tasks have been completed.

Note:

See the notes on Midi2Bin32X.

MIDI Data Format Tables

The MIDI data files processed by the MIDI converter are inherited from the Tone Editor 32X. Thus, judicious control of MIDI on the Tone Editor 32X can improve sound quality. The following describes the control changes and the MIDI formats that can be manipulated using the Tone Editor 32X and that can be beneficial to the production of a sound program using a sequencer.

Rex Sabio
242

MIDI Messages

Message	Function	Comment
\$8n \$xx \$yy	O Note off	Note-off velocity ignored
\$9n \$xx \$yy	O Note on	Velocity 0: note off
\$An \$xx \$yy	X Polyphonic key pressure	
\$Bn \$xx \$yy	O Control change	See table
\$Cn \$xx \$yy	O Program change	Only FM/PSG is valid
\$Dn \$xx \$yy	X Channel pressure	
\$En \$xx \$yy	O Wheel change	Only FM/PSG is valid
\$F0.....\$F7	O System Exclusive	See Table
\$F1-\$F6	X Realtime message	
\$F8-\$FF	X System realtime message	

- The Tone Editor 32X always operates in the omni-off and polyphonic modes.

Rex 242

Control Change

Message	Function	Comment
\$Bn \$01 \$xx	Table vibrato on/off	xx: OFF=\$00-\$3F, ON=\$40-\$7F
\$Bn \$05 \$xx	Sets portamento time at xx/60 seconds.	Transparent to on/off
\$Bn \$07 \$xx	Channel volume	
\$Bn \$0A \$xx	Pan-pot (valid for FM/PWM/Qsound)	For an FM sound source -- \$00-\$29: left, \$55-\$57F: right, \$2A-\$54: center
\$Bn \$10 \$xx	Transpose (CMBIAS)	Halfnote value (xx-\$40)
\$Bn \$11 \$xx	Sets the table vibrato to xx	Does not set vibrato on/off
\$Bn \$14 \$xx	Relative channel volume	Increment: xx-64
\$Bn \$15 \$0x	Specifies the FM sound hard source LFO speed	x: \$00-\$07
\$Bn \$16 \$0x	Specifies the FM sound source hard LFO AME	When the bit N+1 of x is 1, the AME of the operator N is on; otherwise, it is off
\$Bn \$17 \$0x	Specifies the FM sound source hard LFO AMS	x: \$00-\$03
\$Bn \$18 \$0x	Specifies the FM sound source hard LFO PMS	x: \$00-\$07
\$Bn \$19 \$0x	Specifies the FM sound source hard LFO ON/OFF	xx: OFF=\$00-\$3F, ON=\$40-\$7F
\$Bn \$1A \$0x	FM sound source after-touch change of tone ON/OFF	When the bit N+1 of x is 1, the after-touch change of tone for the operator N is on; otherwise, it is off
\$Bn \$1B \$xx	TFlag setting	Defined on a project-by- project basis
\$Bn \$1C \$0x	PSG noise-mode setting	x: \$00-\$07
\$Bn \$1D \$xx	Q Sound on/off	xx: OFF=\$00-\$3F, ON=\$40-\$7F
\$Bn \$41 \$xx	Portamento on/off	xx: OFF=\$00-\$3F, ON=\$40-\$7F
\$Bn \$5E \$xx	Detuning for the (xx-\$40)/N halfnote (FDT)	The value of N depends on the MIDI setting
\$Bn \$79 \$0x	The following items are subject to Reset Control: - Detuning, and transposing - Vendor, FVR, auto- panning, and vibrato - Sound effect mode, and channel pressure	These items are provided to facilitate data editing; they should not be used in music proper
\$Bn \$7B \$00	All note off	Same as above

System Exclusive

Auto-panning

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$mm	\$10	\$00	\$p0	\$p1	\$p2	\$p3	\$p4	\$F7					

This is a M5 Z80 Ver. 2.5x-style auto-panning. For details, see the Driver Manual.

\$mm: MIDI Channel-1
\$p0: Pan Number
\$p1: Table Number
\$p2: Start Point
\$p3: Point Limit
\$p4: Length Counter

If \$p0=0, there should not be any conflict in the values \$p1 through \$p4.

Setting a sound-effect mode

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$00	\$10	\$01	\$h0	\$10	\$h1	\$11	\$h2	\$12	\$h3	\$13	\$F7		

This is a 32X Ver. 3.0-style sound-effect mode. The following F numbers are assigned to operators:

Operator 1 FNumber = \$h0 x 128 + \$10
Operator 2 FNumber = \$h1 x 128 + \$11
Operator 3 FNumber = \$h2 x 128 + \$12
Operator 4 FNumber = \$h3 x 128 + \$13

If \$h0 \$10 - \$h31 \$13 are all zero, the sound-effect mode automatically turns off. If there is any non-zero value, the sound-effect mode automatically turns on.

Direct manipulation of an FM sound source

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$00	\$10	\$02	\$aa	\$rh	\$r1	\$dh	\$d1	\$F7					

The System Exclusive directly writes into an FM sound source. The affinity with the driver is not guaranteed. This option can be used only under special circumstances.

\$aa: Target address. When either 0 or 2, the address indicates \$4000 or \$4002, respectively, in Z80 addresses.
\$rh: Register number, high nibble

\$rl: Register number, low nibble
\$dh: Write data, high nibble
\$dl: Write data, low nibble

FVR

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$mm	\$10	\$03	\$dh	\$dl	\$sh	\$sl	\$ah	\$al	\$ch	\$cl	\$F7		

This is a 32X Ver. 3.0-style FVR. For details, see the Driver Manual.

\$mm: MIDI Channel-1
\$dh: Delay, high nibble
\$dl: Delay, low nibble
\$sh: Stay, high nibble
\$sl: Stay, low nibble
\$ah: Signed addition data, high nibble
\$al: Signed addition data, low nibble
\$ch: Limit, high nibble
\$cl: Limit, low nibble

When \$dh-\$cl are all zero, the FVR automatically turns off. If there is any non-zero value, the FVR automatically turns on.

SSG-type envelope

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$mm	\$10	\$04	\$p0	\$p1	\$p2	\$p3	\$F7						

This parameter sets an SSG-type envelope for an FM sound source.

\$mm: MIDI Channel -1
\$p0: Sets Operator 1.
\$p1: Sets Operator 2.
\$p2: Sets Operator 3.
\$p3: Sets Operator 4.

The value to be set is either \$00 or \$08-\$0F. If the value is \$00, the SSG-type envelope is off. Otherwise, the indicated SSG-type envelope is set.

Detuning specification in division ratio units

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$mm	\$10	\$05	\$xx	\$yy	\$F7								

This parameter assigns a detuning value in division ratio units to either an FM sound source or a PSG sound source.

\$mm: MIDI Channel -1
Detuning value = (\$xx*128 + yy) - \$2000

FM sound source wow

+\$00	+\$01	+\$02	+\$03	+\$04	+\$05	+\$06	+\$07	+\$08	+\$09	+\$0A	+\$0B	+\$0C	+\$0D	+\$0E	+\$0F
\$F0	\$7D	\$mm	\$10	\$06	\$0x	\$ee	\$F7								

When the bit N of \$0x is 1, the \$ee-th table envelope is applied to the operator N+1.

\$mm: MIDI Channel-1

Rex Sabio
242